基于智能代理的网络 QoS 管理模型*)

卞正皑　罗军舟
(东南大学计算机科学与工程系　南京 210096)

摘　要　目前基于代理的 QoS 研究成为一个重要的研究方向，但目前提出的模型大都存在一些问题，如协同性差，不具有自适应能力等等。针对前期研究的主要问题，本文首先提出的一个基于智能代理的 QoS 管理实体模型，它利用了智能代理的特点，使分布式 QoS 管理具有智能性和自适应能力，进而能够适应复杂网络情况下的流量拥塞管理；并在这个实体模型基础上对多代理的协同工作模型进行了研究，最后根据上述定义的实体模型和协同工作结构进行了原型系统的实现。

关键词　网络管理，服务质量，智能代理

Intelligent Agent-Based QoS Management Model

BIAN Zhen-Gai　LUO Jun-Zhou
(Department of Computer Science and Engineering, Southeast University, Nanjing 210096)

Abstract　Agent based QoS management has been an approach which distributed QoS management. But there are some problems in the models that have been made, Such as the lack of the self-adapt ability and cooperation. An intelligent agent based QoS management entry model is presented in the paper. It exploits the feature of intelligent agent to make distributed QoS management more flexible so that the management behavior can adapt to the flow congest control in the complex network. Then a mutli-agent coordination model based on the entry model is represented in the paper. At last a prototype system is implemented according to the entry model and coordination model.

Keywords　Network management, Quality of service, Intelligent agent

1　引言

目前对 QoS 的管理主要还是利用手工的静态配置和简单的策略管理，此类方式 QoS 管理存在着不能够灵活调整而且扩展性差的问题。为此代理技术被开始用于 QoS 管理，目前的研究有两个主要方向，一个是将代理与具体协议结合，如文[1]，提出的是将 RSVP 和代理技术结合的方案，文[2]提出了配置 DiffServ 的移动代理，但它仅仅是将移动代理做为一种简单的配置工具，将代理派发到网络中巡游达到网络配置，以弥补现有网络配置手段的不足，显然没有发挥代理自我计划活动的能力；另一种方案是协议无关的研究，如文[3,13]提出了代理针对 QoS 的瓶颈测量和性能监视方案，这篇文章对代理系统进行了初步的分工，将代理分为应用层，网络层和链路层，在每层都进行资源的协商，类似文[2]提出了透明 QoS 支持的代理平台，以代理作为中间件平台，在这方案中代理具有协商和监视的能力，但它只是一种协商和测量工具，并没有主动预测流量的分布行为。在文[12]中基于代理和 DiffServ 的模型，并证明了对于网络服务中根据拓扑预先处理资源访问的算法从性能上优于传统的资源预留算法，这种方法使得代理可以自主地控制对资源的访问。

在本文提出的模型中，结合了现有的 QoS 管理协议(如策略服务，DiffServ 和 IntServ)，并参考了前期的相关研究，提出了一个基本的代理管理模型：让智能代理感知所处网络环境下的流量和网络介质的状态，并将其转化成一定的数学模型加以表述，而这些工作对于用户服务是透明的；通过引入算法使代理具有对数据流分析和预测能力，不断地通过分析特定业务流的历史情况对流量的分布进行预测，并根据要求对设备进行主动配置；通过智能代理的协商能力使代理之间互相协商保证全局数据流传输的最优化；让代理在网络自我迁移灵活地进行 QoS 协商和配置分发。

2　智能 QoS 代理的实体模型

首先我们提出一个智能代理实体的基本模型，在后续的部份中提出的智能代理均建立在这个基本模型基础上，智能代理的工作过程是按照感知，学习，预测，配置这几个基本步骤进行的。除此之外代理之间的协商需要通过通信组件进行，结构如图 1 所示。

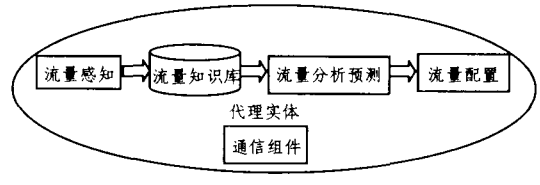

图 1　QoS 智能代理系统结构

2.1　流量的感知过程

流量的感知过程如图 2 所示。智能代理利用 Sniff(嗅探)方式对流量进行被动的感知，同时完成对接口上的数据报文的初步分类。智能代理在设定一个取样时间中根据 IP 五元组识别一个会话过程，获取流量特征参数：平均速率，最大速率和最小速率，最大 MTU 值，包头大小。

智能代理按流量承载的应用层协议对数据流进行分类，分类的目的在于能够直接根据应用协议确定其所需 QoS 指标要求，进而用 Tspec[4] 矢量进行描述，Tspec 矢量中最需要

*)基金项目：国家自然科学基金 90204009，教育部高等学校博士学科点专项科研基金 20030286014。卞正皑　博士研究生，主要研究方向为网络管理等；罗军舟　博士，教授，博士生导师，主要研究方向为协议工程、网络管理、网络安全、网络教育、网络计算等。

关注的是(r,b,p)分量,因为它们直接关系到数据流在设备中的队列时延,抖动以及丢包率等 QoS 指标。

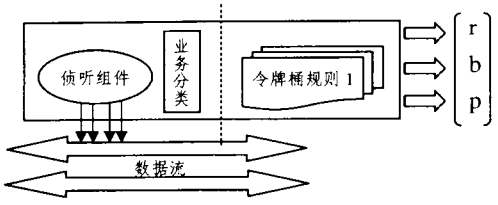

图 2 流量感知过程

在以上工作的基础之上,智能代理将自动生成该数据流的 SLA(Service Level Agreement)。

2.2 智能代理对流量分布规律的学习和预测

智能代理对于流量分布规律的学习包括两方面内容,首先是获知特定应用协议中单个微流在细粒度时间片(秒或分钟)的速率分布规律,分析应用流的瞬时突发特征;另一个是分析边缘节点或流量集中的节点中聚合宏流在粗粒度时间段(小时或天)的周期性分布特征。此外前者的结果经过聚合作为后者的输入,后者根据预定的策略规则对会话微流进行聚合后进行进一步周期性预测,限于篇幅,在文中不再详细说明。

2.3 智能代理对流量的配置

智能代理对流量的配置是通过策略适配器完成,策略适配器将智能代理提交的策略规则映射成相应的管理策略行为,存放到策略库中,由设备进行最后的实施。对于智能代理而言,它所处的是一个异构网络,所以它对网络设备带宽资源的策略规则应是协议无关和设备无关,而具体的 QoS 管理机制则由策略适配器决定。策略适配器将策略规则分成三个部分:策略集,策略行为,策略对象。其中策略集定义了一个设备所包含策略行为的集合,它是整个策略规则的最高层,策略集之间可以互相组合和嵌套,几个策略集可以合在一起成为另一个策略集的子集,高层的策略较低层的策略有更高的优先级;策略行为是策略集的元素,它定义了对被管理对象所实施的配置行为,一个策略集中有多个相关的策略行为;策略对象定义了被管理数据流的特征,凡适合该过滤条件的数据流将被视为被管理对象。

3 多代理的 QoS 协商过程

3.1 代理协作四要素

分布式网络的 QoS 管理需要多代理的协同工作,不同的代理给予不同分工,并为一个共同的目标进行协同工作,对此可以建立一个四元组模型描述即为:

〈agents,goals,knowledgebase,relations〉

Agents 为代理集合,每个代理具有一定解决问题的能力,结合这个模型,我们把这个系统视为一个可自主运行的代理集合,即 agents。

Goals 描述所需完成共同目标任务。每个代理在收集本地的状态信息后,将对带宽资源的要求通过协商转化成一个目标,即 goal。源代理将任务提交给相应的代理集,代理集将任务转换成具体的设备配置并加以实施。

Knowledgebase 代理知识库,它描述了 QoS 管理中各要素的状态以及彼此之间的关系,这部份内容可参阅文[5]。

Relations 不同代理之间的协作关系,在这里,我们根据任务的不同,将代理协同组中的角色分为终端代理、边缘代理、分区代理、协商代理。每种代理在不同的区域活动并完成不同的任务。

3.2 QoS 协商中的任务实体

协商中的任务实体是源代理向目的代理提交的工作,它包括强制性任务和协商性任务;前者是源代理直接指定所需要的带宽资源和其它 QoS 参数,目的代理必须按照任务的要求进行资源分配;后者中源代理只给出一些 QoS 指标要求和相应的松弛度,目的代理可以在松弛度范围内对资源的预留进行调整。

TaskEntry 任务实体的定义包括如下内容:

{TaskID 任务标识号,任务实体描述,任务优先级,QoS 请求分量 1,QoS 请求分量 2……}

任务实体描述设定了被管理对象的条件。它包括以下几个类:主机、端口、网络、链路封装形式,根据对该类的定义目的代理。

任务的 QoS 请求分量,该分量表述了任务实体对目的代理的不同 QoS 指标要求,在一个任务实体中会有若干个请求分量,如下面定义:

Vector1 Delay 10ms,Vector2 Jitter 5ms

这个描述就是要求目的代理必须将队列时延控制在 10ms,抖动控制在 5ms 范围以内。对于 QoS 分量的描述可参阅文[6],对于任务的协同操作需要对任务进行分解,为各相关代理分解生成各子任务,因此每个任务分量需要进行相应分解,如最大路径队列时延、最大抖动和丢包率按照沿路径累加算法;而路径 MTU 和最大可用带宽则需要沿路径的最小化操作。

3.3 资源协商的操作原语

在这里我们定义了代理对资源协商的操作原语,它包括资源请求原语,再协商请求原语,资源释放请求原语,监视请求原语。原语通过 KQML[7] 进行表述,并将 QoS 任务语义要素进行了封装。

(1)资源请求:

源代理向目的代理提出资源预占请求,目的代理根据自身的资源使用情况作出决定并响应源代理的请求,描述为:

代理 A 向代理 B 发送的资源请求原语:

```
ask-if
:content(addReserveval(freebandwidth)=50k
  for (srchost=10.10.138.155))
  :language KIF
  :ontology QoSagents
  :reply-withq1
  :sender agenta
  :receiver agentb
```

代理 B 对代理 A 的请求的确认响应,利用 reply 原语:

```
reply
  : content ( confirmval ( freebandwidth ) = 50k  for ( srchost =
10.10.138.155))
  :language KIF
  :ontology QoSagents
  :in-reply-toq1
  :sender agentb
  :receiver agenta
```

(2)再协商请求:当源节点请求的资源发生变化,通过再协商原语进行资源调整的请求。它的操作原语基本与协商请求相同,只是在 content 参数中有差异。

(3)资源释放请求:源代理通知目的节点代理释放被占用的资源。

(4)监视请求:源代理要求目的代理对特定数据流状态进行监视。当数据流状态发生变化时,及时向源代理发出通告。

3.4 任务协商过程

根据多个代理之间的操作过程,我们将任务协商分为以下几类:

串行协商 代理之间的协商操作是一个串行的过程。这

个过程类似于 RSVP 的 PATH 消息的传递，同样在串行协商过程中，每个代理根据本地节点的资源使用情况对任务进行修正后，将任务提交给下一级代理。但它的协商过程延时长，而且中间节点协商失败就会导致整个协商过程无效。

并行协商 在并行协商过程中，源代理直接与端到端过程中的多个代理进行并发的协商，可以看到源代理并发与沿途的多个代理进行协商，源代理将 QoS 任务分成若干个子任务，同时交给多个代理，由于各代理之间的任务协商是并发执行的，所以这种方式时延小，另外，代理在协商过程中可以直接进行各任务子集的调整。

聚合协商和区分协商 聚合协商即代理能够将不同 QoS 任务聚合成一个新的 QoS 任务，区分协商恰恰相反，它是将一个 QoS 管理任务按照一定规则分成不同的子任务。聚合协商和区分协商的任务合并和分割以令牌桶的计算为基础，这种方式主要用于边缘节点的代理协商。

3.5 代理系统的角色分配

根据系统分工不同，我们定义终端代理、边缘代理、分区代理和协商代理。

终端代理 驻留于终端节点，它的作用包括：

(1)监视节点的每个会话微流，得到会话微流的统计时序分布特性。

(2)对下一个时间片的流量进行预测，将预测结果提交边缘代理。

(3)终端代理还是任务的最终实施者，它通过与分区代理进行协商后，将协商的任务转换成相应的策略，最后再转化成设备的具体配置。

边缘代理 驻留在边缘路由，它的作用包括：

(1)根据历史统计对聚合流的下一个时间周期进行预测，进行初步配置。

(2)对终端代理提交的微流预测值进行聚合，在上一步基础上对资源的配置进行微调。

边缘代理所操作的数据流对象是针对聚合流的，由于聚合流数据流量大，不可能对每个会话流进行分析，首先对各业务聚合流时间周期内流量分布趋势的分析，边缘节点首先能够预测下一个时间段内流量分布的大致特征，预先为每种业务流分配所需带宽；并依据各相关终端代理提交的细粒度时间内的流量预测结果对前期预测进行调整。

分区代理 分区代理负责对分区内的 QoS 任务进行最优化分配，并监督任务的实施，分区代理的主要工作过程已经在前面的任务分派和资源协商详细叙述，限于篇幅，这里不再详细讨论。

协商代理 协商代理是为了提高分区内 QoS 的协商效率，在此我们利用移动代理加以实现，一方面它能在对网络带宽资源使用情况进行监视，并预先发现网络中存在的瓶颈节点，它首先在网络中进行遍历，根据每个节点的拓扑状态和数据流的分布特点，判断该节点是否可能存在瓶颈。与此同时，将设备的逐跳路由项与会话流对应起来，形成端到端的显式路由，根据显式路由中的瓶颈节点制订协商代理的巡行计划。除此之外，协商代理能够根据相关代理的意图主动将配置分发到网络各代理，提高 QoS 的配置效率。此外协商代理在复杂 QoS 协商中也有很明显的优势，一个协商代理可以同时携带多个 QoS 协商任务，同时进行多个数据流的协商任务；而在大规模的网络拓扑中，一个 QoS 协商任务可分到不同的协商代理加以完成。

4 原型系统的设计与实现

在我们设计的原型系统利用了三套软件，代理平台选择了 IBM 的 Aglet2.0 平台[8]，流量感知器使用了 JPCAP[9]，流量控制使用的是 TC 套件中的 CBQ 组件[10]，整个系统在 Linux7.2 上实现。

4.1 智能代理基类定义

构造的智能代理的基类继承了 com.ibm.aglet.Aglet，利用了它的 onCreation 方法对代理进行初始化，代理通过 AgletProxy 进行代理相互访问，代理间消息的处理利用了 handleMessage 方法。KQML 语言表述通过 JKQML 类包。我们建立了 getAction()方法和 setAction()方法，这两个方法是为了让代理能够访问外部信息，并对外部环境进行动作，这两个方法本身并不包含对外界的具体访问操作，它是通过将要操作的参数封装到 Message 参数中，传递给执行环境，由执行环境负责具体的操作。

4.2 流量感知器

我们利用 Windows2000 系统的 MediaServer 播放 MPEG 视频流，通过智能代理对所产生数据流进行采样分析和计算，得到的采样如图 3 所示。

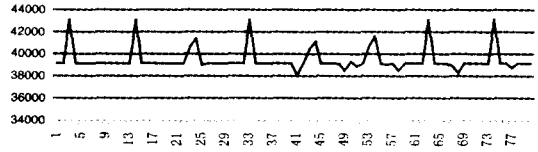

图 3 代理对视频流采样值

基于以上的样值，设定数据流的最大时延为 250 毫秒，最大抖动为 10 毫秒。则可得令牌桶的速度范围如图 4 所示。最终代理可得到该数据流的 SLA 的 Tspec。

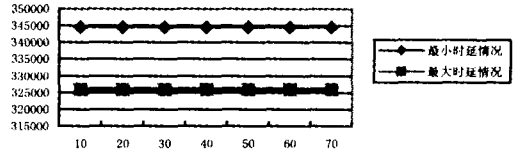

图 4 代理计算的令牌桶速度范围

根据以上的结果，终端代理可制订一个 SLA 协议。

4.3 协商代理

协商代理实现了 MobilityListener 接口，它具有自我迁移能力，除此之外，协商代理类内有两个重要属性，一个为迁移计划 SeqPlanItinerary，另一个为 Task 属性列表；前者定义了协商代理所遍历的节点和在节点所执行的 Task；后者则定义了 Task 属性列表的具体管理内容。

在实现的原型系统中，提交给协商代理两个任务，一个是通过流量感知器对本地会话数据流进行监视，协商代理的 filter 属性定义了所需要监视的数据流对象；另一个任务是负责遍历会话路径的相关节点，与本地代理进行任务协商并进行配置的分发。

4.4 对聚合流量的分析预测

在实验中，边缘代理采集了 30 天中实验室服务器每小时聚合数据流总计，通过 BP 神经网络进行离线分析，对我们的算法进行验证，在计算中把前 20 天的数据作为网络的训

(下转第 36 页)

3 扩展 LSP 的环路消除技术

由 2.2 节分析知，由于文[12]给出的 LSP 扩展机制采用逐个 FA 扩展的方式来扩展 LSP，即当移动节点由 FA1 经 FA2，FA3，…，FAx－1 移动到 FAx 时，它首先将 LSP 由 FA1 扩展到 FA2，再由 FA2 扩展的 FA3，以其逐跳扩展，直到 FAx，并将扩展后的 x－1 条 LSP 简单相加作为 FA1 到 FAx 的 LSP，由于在相加过程没有采用任何环路消除技术，因此，FA1 到 FAx 的 LSP 可能存在环路。

为了消除扩展 LSP 的环路，我们对文[12]给出的 LSP 扩展机制作如下修改：假定移动节点决定从 FA1 开始采用 LSP 扩展机制，则当移动节点经 FA2，FA3，…，FAx－1 移动到外地代理 FAx 时，扩展的 LSP 总是从 FA1 开始建立，即 FAx 总是将绑定更新消息发到 FA1，再由 FA1 发送标签请求消息到 FAx，FAx 在收到标签请求消息后向 FA1 回送标签映射消息，从而建立 FA1 到 FAx 的扩展 LSP。由于 LSP 的建立采用 LDP[13] 协议，而 LDP 协议本身具有环路检测能力，因此，利用 LDP 的环路检测能力，我们便能避免在 FA1 和 FAx 间的扩展 LSP 存在环路。

为了说明修改后的 LDP 扩展机制的工作原理，我们以图 3 为例。当移动节点由 LER/FA1 移动到 LER/FA2 再到 LER/FA3 后，将由 LER/FA1 发送标签请求消息到 LER/FA3，直接建立到 LER/FA3 的扩展 LSP，而非先由 LER/FA2 发送标签请求消息，建立到 LER/FA3 的 LSP 后，再将 LER/FA1 到 LER/FA2 的 LSP 与 LER/FA2 到 LER/FA3 的 LSP 简单相加，作为 LER/FA1 到 LER/FA3 之间的扩展 LSP。

结束语 LSP 扩展机制能有效减小 MPLS 与移动 IP 结合时的信令开销、切换时延和切换丢包，但其扩展的 LSP 可能存在环路。针对该问题，本文给出了一种新的 LSP 扩展的环路消除技术，以避免在扩展的 LSP 上出现环路，从而达到节省网络资源，减小分组的时延的目的。

进一步的研究工作是分析扩展 LSP 环路消除机制的性能。

参 考 文 献

1 Perkins C. IP mobility support. IETF RFC 3344, 2002
2 Gustafsson E, Jonsson A, Perkins C E. Mobile IPv4 regional registration. IETF INTERNET DRAFT, 2003
3 Perkins C, Johnson D B. Route optimization in mobile IP. IETF INTERNET DRAFT draft-ietf-mobileip-optim-09. txt
4 Rosen E, Viswanathan A, Callon R. Multiprotocol label switching architecture. IETF RFC3031. 2001
5 Zhong R, Tham C K, Foo C K, et al. Integration of mobile IP and multi-protocol label switching. In: Proc. IEEE ICC 2001, Helsinki, Finland, 2001
6 Yang T, Makrakis D. Hierarchical mobile MPLS: supporting delay sensitive applications over wireless internet. Intl. Conf. on Info-tech & Info-net (ICII 2001), Beijing, China, 2001
7 Xie QunYing, et al. Handover supporting QoS in MPLS-based hierarchical mobile IPv6 networks. IEEE 58th Vehicular Technology Conf. 2003, 5: 3523～3526
8 Choi J K, Lee Y K, Yang S H, et al. Extension of LDP for mobile IP service through the MPLS Network. IETF INTERNET DRAFT draft-choi-mobileip- ldpext-01. txt, 2001
9 Kim H, Wong K-S D, Chen W, Lau C L. Mobility-aware MPLS in IP-based wireless access networks. IEEE Global Telecommunications Conf. no. 1, 2001. 3444～3448
10 Braden R, Clark D, Shenker S. Integrated services architecture. Internet Engineering Task IETF RFC1633, 1994
11 Blake S, Black D, Carlson M, et al. An architecture for differentiated services. IETF RFC2475, 1998
12 ITU Draft Recommendation Y. MIPoMPLS. Mobile IP services over MPLS. 2003
13 Andersson L, Doolan P, Feldman N, et al. LDP Specification. IETF RFC 3036, 2001

（上接第 29 页）

练样本，以 5 天作为仿真样本，对数据进行分析。由于直接输入训练样本会造成较大的误差，所以我们先对样本进行预处理，将其按 10 取对数，再归一化处理，我们设定最大的训练次数为 60000，得到测试结果如图 5 所示。

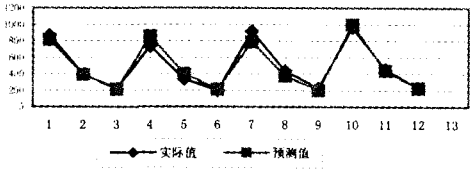

图 5 对聚合数据流的预测值与实际值的比较

从上图可以看到边缘代理可以近似地预测下一时间周期的流量，并主动采取配置行为，这样就可以减小 QoS 协商的时间和次数，减少由于周期性波动而带来的性能降低。

结论 以上叙述了基于智能代理的实体模型和协商结构，并实现了一个基本的原型系统，从上可以看出智能代理与现有的 QoS 协商手段的主要区别有如下几点：首先它是一个协议无关 QoS 协商方法，它与目前研究的 BB[11] 和流量工程的方案不同，它不涉及配置行为的具体协议，这样就减少了由于不同 QoS 方案相互映射的复杂问题；第二代理的协商是非常灵活的，它的协商方式可以是一对一的串行，也可以是一对多的并行和聚合/区分协商，还可以通过移动代理进行复杂 QoS 任务的并行协商。另外代理的 KQML 交互语言可以灵活地扩展新的要素定义，这样就使网络管理任务的分量极易扩展。代理对 QoS 的管理是一种主动的管理，它可以灵活地结合对流量特征的分析和预测预先对资源进行分配，同时还能够对资源的使用进行最优化调整。

本文提出的一个原型系统方案，主要考虑将现有代理平台与目前的 QoS 机制一种整合，但由于 QoS 管理本身和多代理协同工作的复杂性，所以对于多代理条件下的 QoS 分布式感知，协同决策和全局优化仍需要做进一步的研究。

参 考 文 献

1 Sherin m Y. Mohaned A Integrating mobile agents and swarm optimization for efficient qos management in dynamic programable networks pp 358-363[J]. IEEE Melecon 2002
2 Kalaiarul D, Martin C. Transparent Qos Support of Network Applications Using Netlets [A]. In: Proc. of Mobile agents for Telecommunication, Spain October, 2002
3 Manuel G, Torsten B. Internet Service Monitoring with Mobile agents, [J]. IEEEE Network , May/June 2002
4 John W. The Use of RSVP with IETF Integrated Services[EB/OL]. http://www.ietf.org/rfc/rfc2210.txt ,September 1997
5 Schreiber G. 知识工程和知识管理[M]. 北京：机械工业出版社，2003
6 FIPA Quality of Service Ontology Specification[EB/OL], http://www.fipa.org/assets/XC00094A.pdf , 2000
7 Tim M. Draft Specification of the KQML agent-Communication Language[EB/OL], http://www.cs.umbc.edu/kqml/kqmlspec/spec.html ,2000
8 MITSURU O. Aglets Specification 1.1 Draft0.65[EB/OL]. http://aglet.sourceforge.org September, 8th, 1998
9 Keita F. PCAP README[EB/OL] http://netresearch.ics.uci.edu/kfujii/jpcap/README, 2003
10 Netherlabs B. Linux Advanced Routing & Traffic Control HOWTO[EB/OL], http://citeseer.nj.nec.com/correct/20369, January 1998
11 Neilson R. A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment. [EB/OL] http://qbone.internet2.edu/bb/ ,2000
12 Schelén R. Resource Sharing in Advance Reservation Agents[J]. Special issue on Multimedia Networking, 1998, 7(3,4)
13 Pavlou G. QoS management architecture using mobile agents[EB/OL], http://www.ee.surrey.ac.uk/CCSR/ACTS/Miami/mobile-agent-qos.html ,Feb. 2000